

ReST-KV Layer-Wise Eviction Impact on Perplexity and Retrieval in LongBench

Assignee Research

June 9, 2026

Abstract

This report synthesises findings from 8 peer-reviewed papers addressing the following research question: How does ReST-KV's layer-wise eviction strategy influence perplexity and retrieval accuracy in LongBench compared to attention-weight-based methods for sequences longer than 64k tokens. 15 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.2/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: ReST-KV: Robust KV Cache Eviction with Layer-wise Output Reconstruction and Spatial-Temporal Smoothing. Research question: How does ReST-KV's layer-wise eviction strategy influence perplexity and retrieval accuracy in LongBench compared to attention-weight-based methods for sequences longer than 64k tokens?.

2 Methodology

Systematic literature search across multiple databases yielded 8 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 3.2/10.

3 Results

8 papers retrieved. 15 claims extracted; 0 independently verified. Quality review score: 3.2/10.

4 Limitations

This report is a machine-generated literature synthesis and does not constitute original research. Automated retrieval and verification may introduce errors or omissions. Review scores reflect automated assessment, not human peer review. Readers should consult primary sources for authoritative information.

5 Extracted Claims

Claim	Verified	Confidence
ReST-KV is evaluated on five open-source LLMs: Llama2-Chat, Gemma-Instruct, Llama3-Instruct, Mistral-Instruct-v0.3, and	×	0.05
ReST-KV is compared with five baseline methods: StreamingLLM, H2O, TOVA, SnapKV, and LaCache.	×	0.04
ReST-KV is evaluated on four benchmarks: LongBench, RULER, Needle-in-a-Haystack, and InniteBench.	×	0.10
ReST-KV achieves the best performance in most cases on the LongBench benchmark.	×	0.05
ReST-KV reduces peak memory usage by approximately 36.0% compared to full cache at a context length of 128k.	×	0.07
ReST-KV achieves an approximate 10.61 \times speedup over the full cache method at a 128K context length.	×	0.10
ReST-KV is compatible with prell sparse attention approaches, yielding a Time-To-First-Token (TTFT) speedup of up to 3.	×	0.05
ReST-KV uses a xed cache budget to limit the number of KV pairs.	×	0.07
ReST-KV achieves computational complexity comparable to that of SnapKV.	×	0.03
LLMs typically decode text in an auto-regressive manner.	×	0.02
KV cache reduces redundant computation by storing previously computed keys and values.	×	0.06
At each decoding step t , the KV cache stores previously computed keys and values $K_{1:t-1}$, $V_{1:t-1}$ for $X[1 : t - 1]$.	×	0.04
The model requires only the current token x_t to generate x_{t+1} , rather than the full sequence $X = [x_1, \dots, x_t]$.	×	0.02
The query q_t , key k_t , and value v_t are computed as $q_t = x_t W_Q$, $k_t = x_t W_K$, $v_t = x_t W_V$.	×	0.02
The currently computed k_t and v_t will be concatenated with the previously cached keys and values, and used in the attent	×	0.03

References

- <http://arxiv.org/abs/2605.09649v1>
- <http://arxiv.org/abs/2605.08840v1>
- <http://arxiv.org/abs/2606.01563v1>